

# Generalizing Redundancy in Propositional Logic: Foundations and Hitting Sets Duality

Anton Belov<sup>1</sup> and Joao Marques-Silva<sup>1,2</sup>

<sup>1</sup> CASL, University College Dublin, Ireland

<sup>2</sup> IST/INESC-ID, Lisbon, Portugal

**Abstract.** Detection and elimination of redundant clauses from propositional formulas in Conjunctive Normal Form (CNF) is a fundamental problem with numerous application domains, including AI, and has been the subject of extensive research. Moreover, a number of recent applications motivated various extensions of this problem. For example, unsatisfiable formulas partitioned into disjoint subsets of clauses (so-called *groups*) often need to be simplified by removing redundant groups, or may contain redundant *variables*, rather than clauses. In this report we present a generalized theoretical framework of *labelled CNF formulas* that unifies various extensions of the redundancy detection and removal problem and allows to derive a number of results that subsume and extend previous work. The follow-up reports contain a number of additional theoretical results and algorithms for various computational problems in the context of the proposed framework.

## 1 Introduction

Propositional logic formulas in Conjunctive Normal Form (CNF) often have redundant clauses. In some contexts, redundancy is desirable. For example, the identification of redundant clauses is a hallmark of modern SAT solvers [30]. In other contexts, redundancy is undesirable. For example, elimination of redundant clauses is useful in simplifying knowledge bases [24]. A special case of redundancy deals with unsatisfiable subformulas, since the identification of Minimal Unsatisfiable Subformulas (MUSes) finds a wide range of practical applications.

Redundancy in logic has been extensively studied in the recent past [8,24,15,25,26], and includes complexity characterizations of different computational problems. Similarly, the specific case of unsatisfiable subformulas has also been extensively studied [19,22,21]. Computational problems of interest include computing a minimal unsatisfiable subformula, or enumerating them all, and computing an irredundant (or minimal equivalent) subformula, or enumerating them all. Some of these problems have been studied in detail for the case where minimality is expressed in terms of clauses. Moreover, and also for the case where minimality is expressed in terms of clauses, well-known hitting set properties relating minimal unsatisfiable and maximal satisfiable subformulas have been developed for unsatisfiable formulas [32,7,19]. Recently, this work has been extended to the case of satisfiable formulas [21].

---

\* This work is partially supported by SFI grant BEACON (09/IN.1/I2618), and by FCT grants ATTEST (CMU-PT/ELE/0009/2009) and POLARIS (PTDC/EIA-CCO/123051/2010).

Motivated by practical applications, the extraction of MUSes has recently been generalized to groups of (related) clauses [27,31], and to variables [11,12,13,14]. In many settings [27,31], it is important to aggregate related clauses (as *groups* of clauses). In these cases, MUSes need to be expressed in terms of groups of clauses and not in terms of individual clauses. Clearly, MUS problems over groups of clauses or over variables can be extended to the more general case of redundancy removal. For example, one may want to compute a subformula that has no redundant variables, or a subformula that has no redundant groups of clauses. Also relevant are enumeration problems for unsatisfiability and redundancy problems when these problems are expressed in terms of variables or groups of clauses. For example, one may want to enumerate all the variable MUSes of a formula, or all the irredundant subformulas when a problem is represented as groups of (related) clauses.

The main objective of this report is to develop a theoretical framework that provides a unified approach for tackling redundancy problems in CNF formulas, and includes unsatisfiable formulas as a special case. This framework enables the generalization of known theoretical results, but also serves to highlight how existing algorithms for different computational problems can be adapted and extended [29,5,3]. The framework is based on the concept of *labelled CNF* formula, where labels are used to associate individual clauses of a CNF formula with disjoint groups of clauses, or with variables, or with literals, or even with arbitrary intersecting groups of clauses. By extending to the labelled CNF setting the standard definitions of MUSes and MSSes over clauses, the report shows that well-known properties of hitting set duality [32,19,7] also hold for the general case of unsatisfiable labelled CNF formulas, and so hold for MUS and MSS problems over variables, literals or arbitrary groups of clauses. More interestingly, these results also hold for redundancy removal problems for satisfiable formulas, when defined over clauses, variables, or groups of clauses. The immediate consequences of these results include the ability to enumerate MSSes and MUSes of labelled CNF formulas, their extensions to the redundancy removal case, but also the ability to generalize existing MUS extraction algorithms. A detailed description of the report's contributions is included in Section 2 and summarized in Table 2.1.

## 2 Background and Motivation

We focus on formulas in CNF (*formulas*, from hence on), which we treat as finite multi-sets of clauses. We assume that clauses do not contain duplicate variables. Given a formula  $\mathcal{F}$  we denote the set of variables that occur in  $\mathcal{F}$  by  $Var(\mathcal{F})$ , and the set of variables that occur in a clause  $c \in \mathcal{F}$  by  $Var(c)$ . An *assignment*  $\tau$  for  $\mathcal{F}$  is a map  $\tau : Var(\mathcal{F}) \rightarrow \{0, 1\}$ . Assignments are extended to clauses and formulas according to the semantics of classical propositional logic. If  $\tau(\mathcal{F}) = 1$ , then  $\tau$  is a *model* of  $\mathcal{F}$ . If a formula  $\mathcal{F}$  has (resp. does not have) a model, then  $\mathcal{F}$  is *satisfiable* (resp. *unsatisfiable*). By SAT (resp. UNSAT) we denote the set of all satisfiable (resp. unsatisfiable) CNF formulas. Formula  $\mathcal{F}_1$  *implies* formula  $\mathcal{F}_2$  ( $\mathcal{F}_1 \models \mathcal{F}_2$ ) if every model of  $\mathcal{F}_1$  is a model of  $\mathcal{F}_2$ .  $\mathcal{F}_1$  is *equivalent* to  $\mathcal{F}_2$  ( $\mathcal{F}_1 \equiv \mathcal{F}_2$ ) if they have the same set of models. A clause  $c \in \mathcal{F}$  is *redundant* in  $\mathcal{F}$  if  $\mathcal{F} \setminus \{c\} \equiv \mathcal{F}$ , or, equivalently,  $\mathcal{F} \setminus \{c\} \models \{c\}$ . Formulas with (resp. without) redundant clauses are called *redundant* (resp. *irredundant*).

The majority of the research on redundancy in propositional logic addresses unsatisfiable CNF formulas. Irredundant unsatisfiable formulas are called *minimally unsatisfiable* (MU). Explicitly, a formula  $\mathcal{F}$  is MU if (i)  $\mathcal{F} \in \text{UNSAT}$ , and (ii) for any clause  $c \in \mathcal{F}$ ,  $\mathcal{F} \setminus \{c\} \in \text{SAT}$ . A subformula  $\mathcal{F}' \subseteq \mathcal{F}$  is a *minimally unsatisfiable subformula* (MUS) of  $\mathcal{F}$  if  $\mathcal{F}'$  is minimally unsatisfiable. The set of all MUSes of  $\mathcal{F}$  is denoted by  $\text{MUS}(\mathcal{F})$  — in general, a given unsatisfiable  $\mathcal{F}$  may have more than one MUS. MUSes are of interest for a number of reasons, and have been on the radar of AI community for a long time. For example, in early work of Reiter on model-based diagnosis [32], MUSes, under the name of *minimal conflict sets*, are used in computation of a faulty set of components of mis-behaving systems. More recently, MUSes find numerous applications in formal verification of hardware and software systems, product configuration, etc. — see [28] for concrete examples. Motivated by several applications, minimal unsatisfiability and related concepts have been extended to CNF formulas where clauses are partitioned into disjoint sets called *groups* [27,31].

**Definition 1 (Group-Oriented MUS).** *Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_n$ , a group oriented MUS (or, group-MUS) of  $\mathcal{F}$  is a set of groups  $\{\mathcal{G}_{i_1}, \dots, \mathcal{G}_{i_k}\}$ ,  $i_j > 0$ , such that  $\mathcal{F}' = \mathcal{G}_0 \cup \mathcal{G}_{i_1} \cup \dots \cup \mathcal{G}_{i_k} \in \text{UNSAT}$ , and for every  $1 \leq j \leq k$ ,  $\mathcal{F}' \setminus \mathcal{G}_{i_j} \in \text{SAT}$ .*

Note the special role of group  $\mathcal{G}_0$  (*group-0*) — this group consists of “background” clauses that are included in every group-MUS; because of group-0 a group-MUS, as opposed to MUS, can be empty. In addition to clauses and groups of clauses, minimal unsatisfiability has been defined and analysed in terms of the *variables* of the formula [11,14]. Given a CNF formula  $\mathcal{F}$ , and  $V \subseteq \text{Var}(\mathcal{F})$ , the subformula of  $\mathcal{F}$  *induced* by  $V$  is the formula  $\mathcal{F}|_V = \{c \in \mathcal{F} \mid \text{Var}(c) \subseteq V\}$ . Then,  $\mathcal{F}$  is *variable minimally unsatisfiable* (VMU) if  $\mathcal{F} \in \text{UNSAT}$ , and for any  $V \subset \text{Var}(\mathcal{F})$ ,  $\mathcal{F}|_V \in \text{SAT}$ , i.e. no variable can be removed from the formula without making it satisfiable. Here “removal of a variable” means removal of all clauses that have this variable. *Variable MUSes* (VMUSes) are defined accordingly:  $V \subseteq \text{Var}(\mathcal{F})$  is a VMUS of  $\mathcal{F}$  if  $\mathcal{F}|_V$  is VMU. In [3] variable minimal unsatisfiability has been extended in a number of ways akin to the extension of MUSes with group-MUSes.

A notion dual to minimal unsatisfiability is that of maximal satisfiability: a subformula  $\mathcal{F}' \subseteq \mathcal{F}$  is a *maximally satisfiable subformula* (MSS) of  $\mathcal{F}$  if  $\mathcal{F}' \in \text{SAT}$  and  $\forall c \in \mathcal{F} \setminus \mathcal{F}', \mathcal{F}' \cup \{c\} \in \text{UNSAT}$ . The set of MSSes of a CNF formula  $\mathcal{F}$  is denoted by  $\text{MSS}(\mathcal{F})$ . MSSes are also of much interest in the context of AI. For once, given that an MSS constitutes a maximally consistent part of an inconsistent (i.e. unsatisfiable) formula, MSSes can be used for reasoning in the presence of inconsistency — see [7] for an example of an MSS-based framework for reasoning with inconsistent knowledge. Furthermore, an MSS of maximum cardinality constitutes a set of clauses satisfied by a solution to the Maximum Satisfiability (MaxSAT) problem: given a formula  $\mathcal{F}$  find an assignment that satisfies the maximum number of clauses of  $\mathcal{F}$ .

Given an MSS  $\mathcal{S}$  of  $\mathcal{F}$ , one may also consider a subformula  $\mathcal{F} \setminus \mathcal{S}$  of  $\mathcal{F}$  — such subformula is called a *co-MSS* of  $\mathcal{F}$ , and the set of all co-MSSes of  $\mathcal{F}$  is denoted by  $\text{coMSS}(\mathcal{F})$ . Note that when  $\mathcal{F} \in \text{UNSAT}$ , a co-MSS of  $\mathcal{F}$  is a minimal subformula of  $\mathcal{F}$ , removal of which from  $\mathcal{F}$  will regain its satisfiability. Thus, for example, in the context of Reiter’s model-based diagnosis framework [32], co-MSSes constitute the

**Table 2.1.** Summary of existing work on redundancy in CNF formulas. The framework of *labelled CNF formulas* proposed in this report allows to “cover” all the empty entries.

Problem		Clauses	Groups	Variables
MUS/MSS/coMSS		[16,10,14,28]	[27,31]	[11,14,3]
MES/MNS/coMNS		[24,22,21]		
Hitting Set Theorem	UNSAT	[32,7,9,20,2]		
	SAT	[21]		
MaxSAT (algorithms)		[23,1,17]	[18]	

minimal set of components of the faulty system that must be removed to restore its correct behaviour, i.e. the *minimal diagnosis*. For a similar reason, in [27] the authors refer to co-MSSes as *minimal correction subsets (MCSEs)*.

The MUSes, MSSes and co-MSSes of a given unsatisfiable formula  $\mathcal{F}$  are connected via so-called *hitting sets duality theorem*. This theorem has been proved and re-proved on a number of occasions, starting with [32], and later in [7,9,2,27]. The connection is expressed in terms of *irreducible hitting sets*.

**Definition 2 ((Irreducible) Hitting Set).** Let  $\mathcal{S}$  be a collection of arbitrary sets. A set  $H$  is called a *hitting set* of  $\mathcal{S}$  if for all  $S \in \mathcal{S}$ ,  $H \cap S \neq \emptyset$ . A *hitting set*  $H$  is *irreducible*, if no  $H' \subset H$  is a hitting set of  $\mathcal{S}$ .

Then, the hitting set duality theorem states that every MUS of a formula  $\mathcal{F}$  is an irreducible hitting set of the set of co-MSSes of  $\mathcal{F}$ , and vice versa.

**Theorem 1 (cf. [32,7,9,2]).** For any unsatisfiable CNF formula  $\mathcal{F}$ : (i) formula  $\mathcal{M}$  is a co-MSS of  $\mathcal{F}$  if and only if  $\mathcal{M}$  is an irreducible hitting set of  $\text{MUS}(\mathcal{F})$ ; (ii) formula  $\mathcal{U}$  is an MUS of  $\mathcal{F}$  if and only if  $\mathcal{U}$  is an irreducible hitting set of  $\text{coMSS}(\mathcal{F})$ .

Besides exposing an interesting connection between the various subformulas of CNF formulas, hitting set duality is used in algorithms for computation of the set of *all* MUSes of CNF formulas — see, for example, [2,27].

The case of redundancy in satisfiable CNF formulas has also been analysed extensively, for example in [24,22,21]. Here the first object of interest is a subformula of a CNF formula  $\mathcal{F}$  that is irredundant and equivalent to  $\mathcal{F}$  — such subformulas are called *minimal equivalent subformulas (MESEs)*: a subformula  $\mathcal{F}' \subseteq \mathcal{F}$  is an MES of  $\mathcal{F}$  if  $\mathcal{F}' \equiv \mathcal{F}$ , and  $\forall c \in \mathcal{F}'$ ,  $\mathcal{F}' \setminus \{c\} \not\equiv \mathcal{F}$ . The set of all MESEs of  $\mathcal{F}$  is denoted by  $\text{MES}(\mathcal{F})$ . A number of efficient algorithms for computation of MESEs have recently been proposed in [4]. The dual notion is that of a *maximal non-equivalent subformula (MNS)*: a subformula  $\mathcal{F}' \subseteq \mathcal{F}$  is an MNS of  $\mathcal{F}$  if  $\mathcal{F}' \not\equiv \mathcal{F}$  and  $\forall c \in \mathcal{F} \setminus \mathcal{F}'$ ,  $\mathcal{F}' \cup \{c\} \equiv \mathcal{F}$ . The set of MNSes of a CNF formula  $\mathcal{F}$  is denoted by  $\text{MNS}(\mathcal{F})$ . Finally, a subformula of  $\mathcal{F}$  that is a complement of some MNS of  $\mathcal{F}$  is called a *co-MNS* of  $\mathcal{F}$ , and the set of all co-MNSes of  $\mathcal{F}$  is denoted by  $\text{coMNS}(\mathcal{F})$ . Note that, as opposed to the case of unsatisfiable formulas, to our knowledge no extensions of MESEs and related concepts, to groups of clauses or to the variables of CNF formulas have been proposed.

Table 2.1 summarizes existing work on redundancy over clauses, groups of clauses and variables. A number of concrete problems and properties can be considered, namely

minimal unsatisfiability, irredundant (or minimal equivalent) subformulas, hitting set duality theorem and maximum satisfiability. The table shows references for overviews or key references for each topic. In the next section we describe a framework of so-called *labelled CNF formulas*. This framework serves to generalize all of the existing work described above, and, in particular, allows to “cover” all of the empty entries in the table. We demonstrate the usefulness of the framework by deriving a generalized version of the hitting set duality theorem. As a by-product we extend the recent results on irredundant formulas for the case of satisfiable formulas [21]. In addition to the problems shown in Table 2.1, the framework of labelled CNFs allows addressing redundancy problems over literals, wire-MUSes for Boolean circuits [6], and *interesting variables MUS* problem [3].

### 3 Generalized Redundancy

#### 3.1 Labelled CNF Formulas

The key observation that motivates the development of the labelled CNF framework is that in all cases described in Section 2 below, the redundancy in a CNF formula  $\mathcal{F}$  can be analyzed in terms of *possibly intersecting* (i.e. not necessarily disjoint) subsets of clauses of  $\mathcal{F}$ . An additional feature of some of the cases, for example group-MUS, is the presence of the background, or group-0, clauses. We capture the semantics of the intersecting and the background subsets of clauses in the following way.

**Definition 3 (Labelled CNF Formula).** Let  $Lbl$  be a non-empty set of clause labels. A labelled CNF (LCNF) formula  $\Phi$  is a tuple  $\langle \mathcal{F}, \lambda \rangle$ , where  $\mathcal{F}$  is a CNF formula, and  $\lambda : \mathcal{F} \rightarrow 2^{Lbl}$  is a (total) labelling function such that for all  $c \in \mathcal{F}$ ,  $\lambda(c)$  is finite.

We refer to the formula  $\mathcal{F}$  as a *CNF part* of  $\Phi$ , and denote it by  $\mathcal{F}_\Phi$ . The labelling function  $\lambda$  of  $\Phi$  is denoted by  $\lambda_\Phi$ . The set of labels  $\lambda_\Phi(c)$  for  $c \in \mathcal{F}_\Phi$  is referred to as a *set of clause labels of  $c$  in  $\Phi$* . For  $l \in Lbl$ , we refer to the set of clauses  $\mathcal{F}_\Phi^l = \{c \in \mathcal{F}_\Phi \mid l \in \lambda_\Phi(c)\}$  as the *set of clauses labelled with  $l$* . The role of labels in LCNF formulas is to group the clauses of the CNF part into subsets — these subsets can be disjoint, as, for example, in group-CNF context [27,31], or intersecting, as in the context of variable-MUS problem [11,14]. By  $\mathcal{F}_\Phi^\emptyset$  we denote the set  $\{c \in \mathcal{F}_\Phi \mid \lambda_\Phi(c) = \emptyset\}$  of *unlabelled clauses*. These clauses play the role of group-0 clauses in group-CNFs, or uninteresting variables in the extensions of variable-MUS problem [3]. The subscripts for the CNF part and the labelling function of  $\Phi$  may be omitted when  $\Phi$  is understood from the context. With a slight abuse of notation, by  $\lambda(\Phi)$  we denote the set of *active labels* of  $\Phi$ , that is the set  $\bigcup_{c \in \mathcal{F}_\Phi} \lambda(c)$ . Note that  $\lambda(\Phi)$  is finite, and may be empty. Some natural examples of labelling functions and labelled CNFs will be given shortly. The (un)satisfiability, models, and all related concepts of propositional logic are defined for labelled CNFs with respect to their CNF part. For example,  $\Phi$  is unsatisfiable ( $\Phi \in \text{UNSAT}$ ), if  $\mathcal{F}_\Phi \in \text{UNSAT}$ .

**Definition 4 (Induced subformula).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula, and let  $L \subseteq \lambda(\Phi)$ . Then, the subformula of  $\Phi$  induced by  $L$ , is a labelled CNF formula  $\Phi|_L = \langle \mathcal{F}|_L, \lambda \rangle$ , where  $\mathcal{F}|_L = \{c \in \mathcal{F} \mid \lambda(c) \subseteq L\}$ .

In other words,  $\Phi|_L$  has the same labelling function  $\lambda$  as  $\Phi$ , however the CNF part of  $\Phi|_L$  contains only those labelled clauses of  $\mathcal{F}$  *all* of whose labels are included in  $L$  and all the unlabelled clauses  $\mathcal{F}$ , i.e.  $\lambda(\Phi|_L) \subseteq L$ . Alternatively, any clause that has some label outside of  $L$  is removed from  $\mathcal{F}$ . Thus, it will be convenient to speak of an operation of *removal* of a label from  $\Phi = \langle \mathcal{F}, \lambda \rangle$ . Let  $l \in \lambda(\Phi)$  be any (active) label, then, the LCNF formula  $\langle \mathcal{F} \setminus \mathcal{F}^l, \lambda \rangle$  will be said to be obtained by the *removal of label  $l$  from  $\Phi$* . Note that Definition 4 implies that for any  $L \subset \lambda(\Phi)$  (note the strict inclusion), we have  $\mathcal{F}_{\Phi|_L} \subset \mathcal{F}_\Phi$ . Also, note that it is possible that  $\lambda(\Phi|_L) \subset L$  — for example, if for some  $l \in \lambda(\Phi) \setminus L$ , and some  $l' \in L$ ,  $\mathcal{F}^{l'} \subseteq \mathcal{F}^l$ , then  $l' \notin \lambda(\Phi|_L)$ .

*Example 1.* Let  $Lbl = \mathbb{N}$ , and let  $\Phi = \langle \{c_1, \dots, c_8\}, \lambda \rangle$  with the clauses  $c_i$  and the labelling function  $\lambda$  defined as follows (the sets of clause labels are shown as subscripts).

$$\begin{aligned} c_1 &= (\neg y)_{\{1\}} & c_3 &= (z \vee t)_{\{1\}} & c_5 &= (x \vee y \vee z)_{\emptyset} & c_7 &= (\neg y \vee t)_{\{3\}} \\ c_2 &= (y \vee \neg t)_{\{1\}} & c_4 &= (\neg x)_{\{1,2\}} & c_6 &= (\neg x \vee y)_{\{2,3\}} & c_8 &= (\neg t)_{\{4\}} \end{aligned}$$

The set of active labels of  $\Phi$  is  $\lambda(\Phi) = \{1, 2, 3, 4\}$ .  $\Phi$  is satisfiable, with the (only) model  $\{\neg x, \neg y, z, \neg t\}$ . The subformula of  $\Phi$  induced by the set of labels  $L = \{2, 3, 4\}$  is  $\Phi|_L = \langle \{c_5, \dots, c_8\}, \lambda \rangle$ . Additional examples of induced subformulas are  $\Phi|_{\{1,4\}} = \langle \{c_1, c_2, c_3, c_5, c_8\}, \lambda \rangle$  and  $\Phi|_{\emptyset} = \langle \{c_5\}, \lambda \rangle$ .

In the context of redundancy removal in CNF formulas, we speak of redundant clauses, and the basic, atomic, operation on CNF formulas consists of a removal of a single clause from the formula. For the general case of labelled CNF formulas the operation of removal of a single clause is not permitted — instead, the atomic modification to labelled CNFs is a removal of a single (active) *label*, that is *all* clauses in the CNF part of the formula that are labelled with this label. This is an *essential* point of the framework proposed in this report. In fact, when we speak of (proper) subformulas of labelled CNF formulas, we *always* mean “subformulas obtained by removal of labels”, or to be precise:  $\Phi'$  is a subformula of  $\Phi$ , if  $\Phi' = \Phi|_L$  for some  $L \subseteq \lambda(\Phi)$ . When the inclusion is strict, i.e.  $L \subset \lambda(\Phi)$ ,  $\Phi'$  is a *proper* subformula of  $\Phi$ . We will use set notation to denote subformula relation, e.g.  $\Phi' \subset \Phi$ . Note that all subformulas of  $\Phi$  have the same set of unlabelled clauses. Finally, we point out that while  $\Phi' \subseteq \Phi$  implies  $\mathcal{F}_{\Phi'} \subseteq \mathcal{F}_\Phi$ , the fact that  $\mathcal{F}' \subseteq \mathcal{F}$  does not necessarily imply  $\langle \mathcal{F}', \lambda \rangle \subseteq \langle \mathcal{F}, \lambda \rangle$  — again, because removal of a single clause is, in general, not allowed in LCNFs.

### 3.2 Redundancy in Labelled CNFs

It is not difficult to see that, similar to the case of (plain) CNF, removal of labels from labelled CNF formula can never *reduce* the set of models of the formulas, that is, when  $\Phi'$  is a subformula of  $\Phi$ , we always have  $\Phi \models \Phi'$ . However, as with CNFs, removal of some labels from  $\Phi$ , might not affect the set of models of  $\Phi$  at all — such labels are then *redundant*, i.e. all clauses that are labelled with such labels can be removed from the formula while preserving the logical equivalence.

**Definition 5 (Redundant label; Redundant LCNF).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A label  $l \in \lambda(\Phi)$  is *redundant* in  $\Phi$  if  $\Phi|_{\lambda(\Phi) \setminus \{l\}} \equiv \Phi$ . A formula  $\Phi$  is *redundant* if  $\lambda(\Phi)$  contains redundant labels.



Alternatively, a label  $l \in \lambda(\Phi)$  is redundant in  $\Phi = \langle \mathcal{F}, \lambda \rangle$  if  $(\mathcal{F} \setminus \mathcal{F}^l) \models \mathcal{F}^l$ . An irredundant LCNF has the property that the removal of any label from it extends the set of its models — when the formula is unsatisfiable, this means that the removal of any label makes it satisfiable, i.e. it is *minimally* unsatisfiable.

**Definition 6 (Minimally Unsatisfiable LCNF).** A labelled CNF formula  $\Phi = \langle \mathcal{F}, \lambda \rangle$  is minimally unsatisfiable if  $\Phi \in \text{UNSAT}$ , and for any  $L \subset \lambda(\Phi)$ ,  $\Phi|_L \in \text{SAT}$ .

The following example demonstrates a number of natural definitions of labelling functions under which redundant labels capture some well-known notions of redundancy (cf. Section 2).

*Example 2.* Let  $\mathcal{F}$  be any CNF formula.

- (i) Take  $\lambda$  to be such that each clause of  $\mathcal{F}$  is labelled with a single distinct label. Then a label  $l$  is redundant in  $\Phi = \langle \mathcal{F}, \lambda \rangle$  if and only if the (only) clause labelled with  $l$  is redundant, in the plain CNF sense, in  $\mathcal{F}$ .
- (ii) Take  $\lambda$  to be such that each clause of  $\mathcal{F}$  is either labelled with a single, but not necessarily distinct label, or unlabelled. Then a label  $l$  is redundant in  $\Phi = \langle \mathcal{F}, \lambda \rangle$  if and only if the set of clauses  $\mathcal{F}^l$  is redundant, and so we capture the semantics of redundant groups in the group-CNF formulas. The unlabelled clauses  $\mathcal{F}^\emptyset$  correspond to group-0.
- (iii) Take  $L_{bl} = \text{Var}(\mathcal{F})$ , and  $\lambda(c) = \text{Var}(c)$  for each  $c \in \mathcal{F}$ . Then, a label  $v$  is redundant in  $\Phi = \langle \mathcal{F}, \lambda \rangle$  if and only if the variable  $v$  is redundant in  $\mathcal{F}$ . Thus, when  $\Phi$  is minimally unsatisfiable,  $\mathcal{F}$  is variable minimally unsatisfiable (VMU).

As with the case of CNF, by iteratively removing redundant labels from LCNF  $\Phi$  we can obtain a subformula  $\Phi'$  of  $\Phi$  that is equivalent to  $\Phi$  and irredundant. Thus, the subformula  $\Phi'$  is a labelled CNF analog of an MES for (plain) CNF formulas (cf. Section 2). However, in our framework we chose to define labelled MESes in terms of subsets of *labels*, rather than subformulas. We argue that this definition is more natural. Consider, for example, the case of variable-MUSes (VMUSes). Here, VMUS is a subset minimal set of *variables* of an unsatisfiable CNF formula, rather than the subformula induced by these variables. If variables are used as labels of clauses in the LCNF framework, as in Example 2(iii), then it is indeed the subset of labels of the formula that we are interested in, and not the subformula itself.

**Definition 7 (Labelled Minimal Equivalent Subset (LMES)).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled minimal equivalent subset (LMES) of  $\Phi$ , if  $\Phi|_L \equiv \Phi$ , and  $\forall L' \subset L$ ,  $\Phi|_{L'} \not\equiv \Phi$ . The set of all LMESes of  $\Phi$  is denoted by  $\text{LMES}(\Phi)$ .

As with (plain) CNF formulas, when  $\Phi$  is unsatisfiable, LMESes of  $\Phi$  capture the generalized notion of minimally unsatisfiable subformulas.

**Definition 8 (Labelled Minimal Unsatisfiable Subset (LMUS)).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled minimal unsatisfiable subset (LMUS) of  $\Phi$ , if  $\Phi|_L \in \text{UNSAT}$ , and  $\forall L' \subset L$ ,  $\Phi|_{L'} \in \text{SAT}$ . The set of all LMUSes of  $\Phi$  is denoted by  $\text{LMUS}(\Phi)$ .

**Table 3.1.** Summary of the corner cases for CNF and LCNF formulas. Here  $\mathcal{F}$  refers to CNF formula,  $\Phi$  to LCNF.

	Exists for every formula ?	Can be empty formula	Can be the whole formula
MES	yes	yes, only when $\mathcal{F} = \emptyset$	yes
LMES	yes	yes, only when $\lambda(\Phi) = \emptyset$ , or when $\mathcal{F}_\Phi^\emptyset \neq \emptyset$ and all labels are redundant	yes
MNS	no: when $\mathcal{F} = \emptyset$	yes	no
LMNS	no: when $\lambda(\Phi) = \emptyset$ , or when $\mathcal{F}_\Phi^\emptyset \neq \emptyset$ and all labels are redundant	yes	no
coMNS	same as MNS	no	yes
coLMNS	same as LMNS	no	yes

To put the above definitions into a concrete context, consider the labelled CNFs discussed in Example 2: for the case (i) the LMEs correspond to CNF-based MESes and LMUSes correspond to MUSes; for the case (ii) the LMUSes correspond to group-MUSes; for the case (iii) the LMUSes correspond to variable-MUSes (VMUSes).

Note that, by definition, when a label  $l$  is irredundant in  $\Phi$ , every LMES of  $\Phi$  must include  $l$ , and, in fact, the set of all irredundant labels of  $\Phi$  is precisely  $\bigcap \text{LMES}(\Phi)$ . Thus,  $\Phi$  is irredundant if and only if  $\text{LMES}(\Phi) = \{\lambda(\Phi)\}$ . Also, note that a label might be redundant in  $\Phi$ , but irredundant in a subformula  $\Phi'$  of  $\Phi$ . However, if  $l$  is irredundant in  $\Phi$ , it is irredundant in every subformula of  $\Phi$ .

Clearly, every labelled CNF formula  $\Phi$  has at least one LMES, and, furthermore, for any subformula  $\Phi'$  of  $\Phi$ ,  $\Phi' \equiv \Phi$  if and only if some LMES of  $\Phi$  is a subset of  $\lambda(\Phi')$ . Note that in case of CNF formulas, an MES can be empty only if the formula itself is empty. For the case of labelled CNFs, an empty LMES can also occur when all labels are redundant — but this can only happen in the presence of unlabelled clauses. Note that this additional case is not an artifact of the LCNF framework, but rather the artifact of the idea of group-0 clauses (in group-CNFs), and uninteresting variables (in the extensions of variable-MUSes). For example, group-MUS is empty when group-0 is unsatisfiable. Table 3.1 contains a summary of this and other corner cases in the LCNF framework, and contrasts them with the corner cases in (plain) CNF redundancy.

*Example 3.* Consider the LCNF formula  $\Phi$  from Example 1, for convenience we reproduce it here.

$$\begin{aligned}
c_1 &= (\neg y)_{\{1\}} & c_3 &= (z \vee t)_{\{1\}} & c_5 &= (x \vee y \vee z)_{\emptyset} & c_7 &= (\neg y \vee t)_{\{3\}} \\
c_2 &= (y \vee \neg t)_{\{1\}} & c_4 &= (\neg x)_{\{1,2\}} & c_6 &= (\neg x \vee y)_{\{2,3\}} & c_8 &= (\neg t)_{\{4\}}
\end{aligned}$$

To aid the understanding of the example note the following: the clauses  $c_1, \dots, c_4$  are implied by the clauses  $c_5, \dots, c_8$  ( $c_1$  is derived from  $c_7, c_8$  by resolution;  $c_2$  is subsumed by  $c_8$ ;  $c_3$  is derived from  $c_5, c_6, c_7$ ;  $c_4$  is derived from  $c_6, c_7, c_8$ ); also, the clauses  $c_6, c_7, c_8$  are implied by the clauses  $c_1, c_2, c_4$  ( $c_6$  is subsumed by  $c_4$ ;  $c_7$  is subsumed by  $c_1$ ;  $c_8$  is derived from  $c_1, c_2$ ).

Label 1 is redundant in  $\Phi$  due to the fact that clauses  $\mathcal{F}^1 = \{c_1, \dots, c_4\}$  are implied by  $\mathcal{F}|_{\{2,3,4\}} = \{c_5, \dots, c_8\}$ . However, labels 2, 3 and 4 are irredundant in  $\Phi|_{\{2,3,4\}}$ ,



hence  $L_1 = \{2, 3, 4\}$  is a labelled MES of  $\Phi$ . The formula  $\Phi$  has another LMES: label 3 is redundant in  $\Phi$ , as clauses  $\mathcal{F}^3 = \{c_6, c_7\}$  are implied by  $\mathcal{F}|_{\{1,2,4\}} = \{c_1, \dots, c_5, c_8\}$ . However,  $\Phi|_{\{1,2,4\}}$  contains a redundant label 4, as clause  $c_8$  is implied by  $c_1, c_2$ . Now,  $\Phi|_{\{1,2\}} = \langle \{c_1, \dots, c_5\}, \lambda \rangle$  is irredundant — even though clause  $c_5$  is implied by  $c_2$  and  $c_3$  and so is redundant in the (plain) CNF sense, we cannot remove it from  $\Phi|_{\{1,2\}}$ ; note that this would also be the case if  $\lambda(c_5) = \{2\}$ . We conclude that  $L_2 = \{1, 2\}$  is an LMES of  $\Phi$ .

The notion dual to minimal equivalence (resp. minimal unsatisfiability) is that of maximal non-equivalence (resp. maximal satisfiability). Here we are interested in sets of labels that induce a subformula of  $\Phi$  that is not equivalent to  $\Phi$ , but an addition of any active label from  $\Phi$ , results in an equivalent subformula.

**Definition 9 (Labelled Maximal Non-equivalent Subset (LMNS)).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled maximal non-equivalent subset (LMNS) of  $\Phi$ , if  $\Phi|_L \not\equiv \Phi$  and for every  $L', L \subset L' \subseteq \lambda(\Phi)$ ,  $\Phi|_{L'} \equiv \Phi$ . The set of all LMNSes of  $\Phi$  is denoted by  $\text{LMNS}(\Phi)$ .

Note that just as with clausal MNSes, which do not exist for empty formulas because every subformula of an empty formula is equivalent to it, LMNSes do not exist for LCNF formulas with  $\lambda(\Phi) = \emptyset$ . Also, just as with LMESes, the presence of unlabelled clauses gives rise to an additional corner case (see also Table 3.1) — when all labels are redundant (for non-empty formulas this can only happen if  $\mathcal{F}^\emptyset \neq \emptyset$ ), every subformula of  $\Phi$  is also equivalent to  $\Phi$ . For the case of unsatisfiable LCNFs, we have a definition analogous to that of (clausal) MSS.

**Definition 10 (Labelled Maximal Satisfiable Subset (LMSS)).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled maximal satisfiable subset (LMSS) of  $\Phi$ , if  $\Phi|_L \in \text{SAT}$  and for every  $L', L \subset L' \subseteq \lambda(\Phi)$ ,  $\Phi|_{L'} \in \text{UNSAT}$ . The set of all LMSSes of  $\Phi$  is denoted by  $\text{LMSS}(\Phi)$ .

Note that as opposed to MSSes, which exist for every CNF formula, LMSSes do not exist for formulas with an unsatisfiable set of unlabelled clauses, because no subformula of such a formula is satisfiable.

As discussed in Section 2, clausal MSSes are of interest for a number of reasons, one of which that an MSS of maximum cardinality is a set of clauses that are true under a solution to MaxSAT problem. With this in mind we can also define a generalized version of MaxSAT problem.

Given an LMSS  $L$  of  $\Phi$ , one may also consider its complement  $\lambda(\Phi) \setminus L$ . When  $\Phi \in \text{SAT}$ , the complement is an empty set, however when  $\Phi \in \text{UNSAT}$ ,  $\lambda(\Phi) \setminus L$  is a minimal set of labels of  $\Phi$ , removal of which from  $\Phi$ , will regain the satisfiability. The corresponding concept in the context of unsatisfiable CNF is that of co-MSS (cf. Section 2). Similar, though less intuitive, concept arises in the case of LMNSes.

**Definition 11 (co-LMNS).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled co-MNS (co-LMNS) of  $\Phi$ , if  $\lambda(\Phi) \setminus L \in \text{LMNS}(\Phi)$ . Or, explicitly, if  $\Phi_{\lambda(\Phi) \setminus L} \not\equiv \Phi$ , and for any  $L' \subset L$ ,  $\Phi_{\lambda(\Phi) \setminus L'} \equiv \Phi$ . The set of all co-LMNSes of  $\Phi$  is denoted by  $\text{coLMNS}(\Phi)$ .

**Definition 12 (co-LMSS).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula. A set of labels  $L \subseteq \lambda(\Phi)$  is a labelled co-MSS (co-LMSS) of  $\Phi$ , if  $\lambda(\Phi) \setminus L \in \text{LMSS}(\Phi)$ . Or, explicitly, if  $\Phi_{\lambda(\Phi) \setminus L} \in \text{SAT}$ , and for any  $L' \subset L$ ,  $\Phi_{\lambda(\Phi) \setminus L'} \in \text{UNSAT}$ . The set of all co-LMSSes of  $\Phi$  is denoted by  $\text{coLMSS}(\Phi)$ .

*Example 4.* Consider again the LCNF formula  $\Phi$  from Example 1. The formula has three LMNSes:  $\{1, 3, 4\}$ ,  $\{2, 3\}$  and  $\{2, 4\}$ , and three corresponding co-LMNSes.

### 3.3 Generalized Hitting Set Duality

As mentioned in Section 2, for a given CNF formula  $\mathcal{F}$ , there is a relationship between the set of MUSes of  $\mathcal{F}$  and the set of co-MSSes of  $\mathcal{F}$ :  $\text{coMSS}(\mathcal{F})$  is a set of irreducible hitting sets of  $\text{MUS}(\mathcal{F})$ . This relationship has been (re)discovered on a number of occasions, with the earliest, to our knowledge, attributed to Reiter [32] in the context of model-based diagnosis — there MUSes are called minimal conflict sets, and coMSSes are called minimal diagnoses. This relationship is a basis for the efficient MUS enumeration algorithms (cf. [2,27]). A weaker form of this relationship, namely  $\bigcup \text{MUS}(\mathcal{F}) = \mathcal{F} \setminus \bigcap \text{MSS}(\mathcal{F})$ , derived by Kullmann [20], has been also generalized in [21] to the case of satisfiable CNF formulas. In this section we develop a general version of the hitting set theorem for the labelled CNF formulas. In addition to subsuming the previous results, the theorem covers all the other, not previously analyzed, cases, e.g. group-MUS or variable-MUS. The theorem also allows to develop effective algorithm computation of the set of all LMESes.

The proof of the theorem relies on a number of basic properties of LMESes and LMNSes, as well as the following known property of irreducible hitting sets (recall Definition 2). The property asserts that every element of an irreducible hitting set must, in a sense, have a “reason” to be there, i.e. to be a unique representative of some set.

**Proposition 1.** Let  $\mathcal{S}$  be a collection of arbitrary sets, and let  $H$  be any hitting set of  $\mathcal{S}$ . Then,  $H$  is irreducible if and only if  $\forall h \in H, \exists S \in \mathcal{S}$  such that  $H \cap S = \{h\}$ .

The hitting sets relationship is captured formally by the following theorem.

**Theorem 2 (Generalized Hitting Set Duality Theorem).** Let  $\Phi = \langle \mathcal{F}, \lambda \rangle$  be a labelled CNF formula, such that  $\lambda(\Phi) \neq \emptyset$ , and if  $\mathcal{F}^\emptyset \neq \emptyset$  then at least one label in  $\lambda(\Phi)$  is irredundant. Then,

- (i)  $L \subseteq \lambda(\Phi)$  is a coLMNS of  $\Phi$  if and only if  $L$  is an irreducible hitting set of  $\text{LMES}(\Phi)$ .
- (ii)  $L \subseteq \lambda(\Phi)$  is an LMES of  $\Phi$  if and only if  $L$  is an irreducible hitting set of  $\text{coLMNS}(\Phi)$ .

Note that the restrictions on the formula  $\Phi$  in the above theorem are in place to ensure that the formula has at least one co-LMNS (cf. Table 3.1). These restrictions are satisfied *a priori* for a number of special cases, which we discuss shortly.

The intuition behind (i) can be explained as follows<sup>3</sup> — since the removal of a co-LMNS from a formula  $\Phi$  makes it non-equivalent to  $\Phi$ , the removal must “break” each

<sup>3</sup> This explanation is a generalized version of the one given for unsatisfiable CNF case in [27]

of the LMESes of the formula. Hence a co-LMNS must include at least one label from each of the LMESes, i.e. it is a hitting set of the set of LMESes of the formula. The minimality of co-LMNS implies the irreducibility of the hitting set, and vice versa.

Before we proceed with the proof of Theorem 2, recall a simple property of sub-formulas of any LCNF formula  $\Phi$  that satisfies the conditions of the theorem: for any  $\Phi' \subseteq \Phi$ ,  $\Phi' \not\equiv \Phi$  if and only if  $\lambda(\Phi')$  is a subset of some LMNS of  $\Phi$ ;  $\Phi' \equiv \Phi$  if and only if  $\lambda(\Phi')$  is a superset of some LMES of  $\Phi$ .

*Proof.* For clarity we adopt the following convention: letter  $S$  will be used to denote LMNSes,  $M$  to denote co-LMNSes,  $U$  to denote LMESes.

*Part (i), If:* Let  $M$  be an irreducible hitting set of  $\text{LMES}(\Phi)$ , and let  $S = \lambda(\Phi) \setminus M$ . First, since  $M$  is a hitting set of  $\text{LMES}(\Phi)$ ,  $S$  cannot include an LMES of  $\Phi$ , and so  $\Phi|_S \not\equiv \Phi$ . Since  $M$  is an *irreducible* hitting set of  $\text{LMES}(\Phi)$ , for any label  $l \in M$ , there exists  $U \in \text{LMES}(\Phi)$ , such that  $M \cap U = \{l\}$  (by Proposition 1). Hence, for any  $l \in M$ , the set  $S \cup \{l\}$  includes some LMES  $U$  of  $\Phi$ , and so  $\Phi|_{S \cup \{l\}} \equiv \Phi$ . We conclude that  $S$  is an LMNS of  $\Phi$ , and so  $M$  is a co-LMNS of  $\Phi$ .

*Part (i), Only-if:* Let  $M$  be any co-LMNS of  $\Phi$ , and let  $S = \lambda(\Phi) \setminus M$  be the corresponding LMNS. Since  $\Phi|_S \not\equiv \Phi$ , for any  $U \in \text{LMES}(\Phi)$ ,  $U \setminus S \neq \emptyset$  (otherwise  $U \subseteq S$ ), and so  $U \cap M \neq \emptyset$ , that is,  $M$  is a hitting set of  $\text{LMES}(\Phi)$ . Now, since  $S$  is an LMNS, for every label  $l \in M$ ,  $\Phi|_{S \cup \{l\}} \equiv \Phi$ . Thus, for every  $l \in M$ , there exists an LMES  $U$  such that  $M \cap U = \{l\}$ . By Proposition 1,  $M$  is an *irreducible* hitting set of  $\text{LMES}(\Phi)$ .

*Part (ii), If:* Let  $U$  be an irreducible hitting set of  $\text{coLMNS}(\Phi)$ . We have that for any  $M \in \text{coLMNS}(\Phi)$ ,  $U \cap M \neq \emptyset$ . Hence, for no  $S \in \text{LMNS}(\Phi)$  we have  $U \subseteq S$  and so  $\Phi|_U \equiv \Phi$ . Since  $U$  is *irreducible*, by Proposition 1, for every label  $l \in U$ , there exists  $M \in \text{coLMNS}(\Phi)$  such that  $U \cap M = \{l\}$ . Thus, for every  $l \in U$ , there exists a co-LMNS  $M$  such that  $U' = U \setminus \{l\} \subseteq \lambda(\Phi) \setminus M$ , i.e.  $U'$  is included in some LMNS of  $\Phi$ , and so  $\Phi|_{U'} \not\equiv \Phi$ . We conclude that  $U \in \text{LMES}(\Phi)$ .

*Part (ii), Only-if:* Let  $U$  be any LMES of  $\Phi$ . Since  $\Phi|_U \equiv \Phi$ ,  $U$  cannot be included in any LMNS of  $\Phi$ , and so for every co-LMNS  $M$  of  $\Phi$ , we have  $U \cap M \neq \emptyset$ , i.e.  $U$  is a hitting set of  $\text{coLMNS}(\Phi)$ . Now, since  $U$  is an LMES of  $\Phi$ , for any label  $l \in U$ ,  $\Phi|_{U \setminus \{l\}} \not\equiv \Phi$ , and so the set  $U \setminus \{l\}$  is included in some LMNS of  $\Phi$ . Hence, for any label  $l \in U$ , there exists a co-LMNS  $M$  of  $\Phi$  such that  $U \cap M = \{l\}$ . Hence, By Proposition 1,  $U$  is an *irreducible* hitting set of  $\text{coLMNS}(\Phi)$ .  $\square$

The restrictions on the formula  $\Phi$  in Theorem 2 can, in some cases, be satisfied *a priori*. Consider, for example, the case  $\Phi \in \text{UNSAT}$ , and the labelling function as in Example 2(i). Since  $\mathcal{F}_\Phi \in \text{UNSAT}$ , we have  $\mathcal{F} \neq \emptyset$ , and every clause is labelled ( $\mathcal{F}^\emptyset = \emptyset$ ), the theorem applies unconditionally to such formulas. Thus, we get exactly the original version of hitting set duality theorem for unsatisfiable CNF formulas (see Section 2). For the case of group-MUS (Example 2(ii)), the theorem holds whenever  $\mathcal{F}^\emptyset \in \text{SAT}$ , as this condition ensures that the formula has at least one irredundant label (since  $\Phi \in \text{UNSAT}$ ).

The following corollary is a straightforward consequence of Theorem 2, and is a generalized version of the relationship between MUSes and co-MSSes shown in [19].

**Corollary 1.** *Let  $\Phi$  be as in Theorem 2. Then,  $\bigcup \text{LMES}(\Phi) = \lambda(\Phi) \setminus \bigcap \text{LMNS}(\Phi)$ .*

The following example illustrates the claims of Theorem 2 and Corollary 1.

*Example 5.* Consider the LCNF formula  $\Phi$  from Example 1. From Examples 3 and 3 we have the following:  $\text{LMES}(\Phi) = \{\{1, 2\}, \{2, 3, 4\}\}$ ,  $\text{LMNS}(\Phi) = \{\{1, 3, 4\}, \{2, 3\}, \{2, 4\}\}$ ,  $\text{coLMNS}(\Phi) = \{\{2\}, \{1, 3\}, \{1, 4\}\}$ . Note that  $\text{LMES}(\Phi)$  has exactly 3 irreducible hitting sets that constitute the set  $\text{coLMNS}(\Phi)$ . Also,  $\bigcup \text{LMES}(\Phi) = \{1, 2, 3, 4\} = \lambda(\Phi)$ , and  $\bigcap \text{LMNS}(\Phi) = \emptyset$ .

## 4 Conclusion

This report presents a framework of labelled CNF formulas that allows to generalize and extend the existing work on redundancy detection and removal in CNF formulas. Future work includes the development of a number of additional theoretical results, and a suite of efficient algorithms that address various computational problems in the context of the proposed framework.

## References

1. C. Ansótegui, M. L. Bonet, and J. Levy. A new algorithm for weighted partial MaxSAT. In *AAAI*. AAAI, 2010.
2. J. Bailey and P. J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *PADL*, pages 174–186, 2005.
3. A. Belov, A. Ivrii, A. Matsliah, and J. Marques-Silva. On efficient computation of variable muses. In *Theory and Applications of Satisfiability Testing SAT 2012*, volume 7317 of *LNCS*, pages 298–311. Springer, 2012.
4. A. Belov, M. Janota, I. Lynce, and J. Marques-Silva. On computing minimal equivalent subformulas. In *CP*, 2012. (to appear).
5. A. Belov and J. Marques-Silva. Accelerating MUS extraction with recursive model rotation. In *FMCAD*, pages 37–40, 2011.
6. A. Belov and J. Marques-Silva. Minimally unsatisfiable Boolean circuits. In *SAT*, pages 145–158, 2011.
7. E. Birnbaum and E. L. Lozinskii. Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.*, 15(1):25–46, 2003.
8. Y. Boufkhad and O. Roussel. Redundancy in random SAT formulas. In *AAAI*, pages 273–278, 2000.
9. R. Bruni. Approximating minimal unsatisfiable subformulae by means of adaptive core search. *Discrete Applied Mathematics*, 130(2):85–100, 2003.
10. H. K. Büning and O. Kullmann. Minimal unsatisfiability and autarkies. In *Handbook of Satisfiability*, pages 339–401. IOS Press, 2009.
11. Z.-Y. Chen and D.-C. Ding. Variable minimal unsatisfiability. In *TAMC*, pages 262–273. Springer-Verlag, 2006.
12. Z.-Y. Chen, Z.-H. Tao, H. Kleine Büning, and L.-F. Wang. Applying variable minimal unsatisfiability in model checking. *Journal of Software*, 19(1):39–47, 2008.
13. C. Desrosiers, P. Galinier, and A. Hertz. Efficient algorithms for finding critical subgraphs. *Discrete Appl. Math.*, 156:244–266, January 2008.
14. C. Desrosiers, P. Galinier, A. Hertz, and S. Paroz. Using heuristics to find minimal unsatisfiable subformulas in satisfiability problems. *J. Comb. Optim.*, 18(2):124–150, 2009.

15. O. Fourdrinoy, É. Grégoire, B. Mazure, and L. Sais. Eliminating redundant clauses in SAT instances. In *CPAIOR*, pages 71–83, 2007.
16. É. Grégoire, B. Mazure, and C. Piette. On approaches to explaining infeasibility of sets of Boolean clauses. In *ICTAI*, pages 74–83, November 2008.
17. F. Heras, A. Morgado, and J. Marques-Silva. Core-guided binary search algorithms for maximum satisfiability. In *AAAI*, 2011.
18. F. Heras, A. Morgado, and J. Marques-Silva. An empirical study of encodings for group MaxSAT. In *Canadian Conference on AI*, 2012.
19. O. Kullmann. An application of matroid theory to the SAT problem. In *CCC*, pages 116–124, 2000.
20. O. Kullmann. Lean clause-sets: generalizations of minimally unsatisfiable clause-sets. *Discrete Applied Mathematics*, 130(2):209–249, 2003.
21. O. Kullmann. Constraint satisfaction problems in clausal form II: Minimal unsatisfiability and conflict structure. *Fundam. Inform.*, 109(1):83–119, 2011.
22. O. Kullmann, I. Lynce, and J. Marques-Silva. Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In *SAT*, pages 22–35, 2006.
23. C. M. Li and F. Manyà. MaxSAT, hard and soft constraints. In *Handbook of Satisfiability*, pages 613–632. IOS Press, 2009.
24. P. Liberatore. Redundancy in logic I: CNF propositional formulae. *Artif. Intell.*, 163(2):203–232, 2005.
25. P. Liberatore. Redundancy in logic II: 2CNF and Horn propositional formulae. *Artif. Intell.*, 172(2-3):265–299, 2008.
26. P. Liberatore. Redundancy in logic III: Non-monotonic reasoning. *Artif. Intell.*, 172(11):1317–1359, 2008.
27. M. H. Liffiton and K. A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008.
28. J. Marques-Silva. Minimal unsatisfiability: Models, algorithms and applications. In *ISMVL*, pages 9–14, 2010.
29. J. Marques-Silva and I. Lynce. On improving MUS extraction algorithms. In *SAT*, pages 159–173, 2011.
30. J. Marques-Silva, I. Lynce, and S. Malik. Conflict-driven clause learning SAT solvers. In *Handbook of Satisfiability*, pages 131–154. IOS Press, 2009.
31. A. Nadel. Boosting minimal unsatisfiable core extraction. In *FMCAD*, pages 121–128, 2010.
32. R. Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.